

Software Systems Verification and Validation

Assoc. Prof. Andreea Vescan

Babeş-Bolyai University

Cluj-Napoca

2019-2020

Lecture 2b: BBT

Software Systems Verification and Validation



Outline

- Testing – fundamental questions
- Black-box testing (domain testing)
 - Equivalence partitioning (EP)
 - Boundary-value analysis (BVA)
 - Advantages/Disadvantages
- Example - black-box testing
 - Example – EP+BVA
- Test management tool – Testlink
- Maven
- Continuous integration tool - Jenkins

Domain testing

- The problem you have to solve
 - You cannot afford to run every possible test.
 - You need a method for choosing a few powerful tests that will represent the rest.
- Domain testing
 - Provides a **sampling strategy**
 - It provides **heuristics** for choosing a small number of tests that are powerful enough to represent the larger domain.
 - Equivalence
 - Boundaries
- Input versus output variables
 - Domain definitions
 - Specification:
 - Data (X) + preconditions
 - Result (Z)+ postcondition
- Primary dimension versus secondary dimension [BBST]
 - Primary dimension - reflects the reason you are entering data into the field
 - Secondary dimension - reflects the other ways the input can vary

Domain testing (Black-box testing)

- Equivalence partitioning
- Boundary-value analysis

Equivalence partitioning

- Equivalence class (EC) - definition [Mye04]
 - a partition of input domain of a program.
- Equivalence partitioning
 - to partition the input domain of a program into a finite number of equivalence classes such that you can reasonably assume that a test of a representative value of each class is equivalent to a test of any other value.

Test-case design by equivalence partitioning

- steps

- Identifying the equivalence classes (EC)
 - Valid equivalence classes.
 - Invalid equivalence classes.
 - Default, empty, blank, null, zero, none.
 - Invalid, wrong, incorrect, garbage data.
- Defining the test cases
 - assign a unique number to each equivalence class;
 - Until all valid/invalid equivalence classes have been covered by (incorporated into) test cases:
 - write a new test case covering as many of the uncovered valid equivalence classes as possible;
 - write a test case that covers one, and only one, of the uncovered invalid equivalence classes.

Test-case design by equivalence partitioning - guidelines

- An input condition specifies a range of values [a,b].
 - ➔ 1 valid EC, 2 invalid EC
- An input condition specifies the number of values “1 to 3 possibilities”.
 - ➔ 1 valid EC and 2 invalid EC
- An input condition specifies a set of input values.
 - ➔ 1 valid EC for each element in the set, 1 invalid EC
- An input condition specifies a must be situation.
 - ➔ 1 valid EC, 1 invalid EC
- If there is any reason to believe that the program does not handle elements in an equivalence class identically, split the equivalence class into smaller equivalence classes.

Boundary-value analysis

- Boundary-value analysis - definition [Mye04]
 - focuses on the boundary areas of a programs input domain
- Boundary conditions
 - Situations directly on, above, and beneath the edges of input EC and output EC.
 - One or more elements should be selected such that each edge of the EC is the subject of a test.
 - BVA explores situations on and around the edges of the EP.

Test-case design by boundary-value analysis -guidelines

- An input condition specifies a range of values $[a,b]$.
 - ➔ the ends of the range, situations just beyond the ends;
- An input condition specifies the number of values “1 to 3 possibilities”.
 - ➔ the minimum and maximum number of values, one beneath and beyond these values;
- An input condition specifies an ordered set of input values.
 - ➔ the first, the last elements of the set;
- The above rules are applied to the output conditions.

Domain testing (Black-box testing)

Advantages

- No knowledge of implementation.
- Tester independent of programmer.
- User's point of view.
- Ambiguities in spec.
- After specifications is completed.

Disadvantages

- A small number of inputs.
- No clear spec.
- Hard to design.
- Unnecessary repetition of test.
- Many program paths untested.
- Specific segments of code?

Example

- **Problem statement:** Compute the number of participants with the maximum score (0 to 100 points possible) at a competition.
- Applied:
 - EP
 - BVA
- See example files
- **Lecture02_Demo**

Maven

- goal - to allow a developer to comprehend the complete state of a development effort in the shortest period of time
- <https://maven.apache.org/what-is-maven.html>
- Maven
 - The Failsafe Plugin is designed to run integration tests.
 - Maven 2 Integration

Testlink

- Test management tool
 - Testlink. (Release 1.9.8)
- <https://www.scs.ubbcluj.ro/testlink>

Jenkins

- Continuous integration tool
- <https://scs.ubbcluj.ro:9090/>

Lecture 2: Required reading (before Seminar 2)

- Assignment 2
Laboratory 2 - BBT
Laboratory 3 - WBT
- Testing – Black-box testing/Domain testing (EC+BVA)
 - In class assignments
 - Homework assignments

References

- [PY08] M. Pezzand and M. Young. *Software Testing and Analysis: Process, Principles and Techniques*. John Wiley and Sons, 2008.
- [Mye04] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004
- [You79] E. Yourdon, *Structured Walkthroughs*, Prentice-Hall, Englewood Cliffs, NJ, 1979
- [Fre10] M. Frentiu, *Verificarea si validarea sistemelor soft*, Presa Universitara Clujeana, 2010
- [BBST] BBST Testing course, <http://testingeducation.org/BBST/>
 - **Test design,**
 - **Lecture 5: Domain testing**
- Tutorials
 - Module Tutorials in canvas

Questions

- Thank You For Your Attention!

Next Lecture

- White-box testing